

PROGRAM PREPARING DEVICE

Patent Number: JP7104977
Publication date: 1995-04-21
Inventor(s): KATSURA SAKITAKE; others:
02
Applicant(s): FUJITSU TEN LTD
Requested Patent: ☒ JP7104977
Application Number: JP19940073621 19940412
Priority Number(s):
IPC Classification: G06F9/06
EC Classification:
Equivalents: JP3046710B2

Abstract

PURPOSE: To efficiently prepare a program and to facilitate a debugging work by making each module have the information related to an arrangement location when a program is completed by combining prepared modules and arranging each module and common parts based on arrangement location related information.

CONSTITUTION: A preparing device 1 reads a standard module 20 from a module library 4a based on a module name 11 of specification extraction data 10 and reads the arrangement related information corresponding to the standard module 20 from an arrangement related information library 4c. A common parts 50 is extracted as the input from type data 14 and the keyboard within specification extraction data 10 is contrasted with the list of the common parts registered in a common parts library 4b and the common parts is arranged as the common parts 83 within a program source 7 in a common parts arrangement part T=73. A map table 40 is arranged as a map table 82 by a map arrangement part 72.

Data supplied from the esp@cenet database - l2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-104977

(43)公開日 平成7年(1995)4月21日

(51) Int.Cl.⁶

G O 6 F 9/06

識別記号

厅内整理番号

F I

技術表示箇所

530

W 9367-5B

Q 9367-5B

審査請求 未請求 請求項の数 4 O L (全 10 頁)

(21)出願番号

特願平6-73621

(22) 出願日

平成6年(1994)4月12日

(31)優先権主張番号

特願平5-199786

(32) 優先日

平 5 (1993) 8 月 11 日

(33)優先權主張国

日本 (JP)

(71)出願人 000237592

富士通テン株式会社

兵庫県神戸市兵庫区御所通1丁目2番28号

(72) 発明者 ▲葛▼ 崎偉

兵庫県神戸市兵庫区御所通1丁目2番28号

富士通テン株式会社内

(72) 發明者 斗納 宏敏

兵庫県神戸市兵庫区御所通1丁目2番28号

富士通テン株式会社内

(72)発明者 細川 健司

兵庫県神戸市兵庫区御所通1丁目2番28号

富士通テン株式会社内

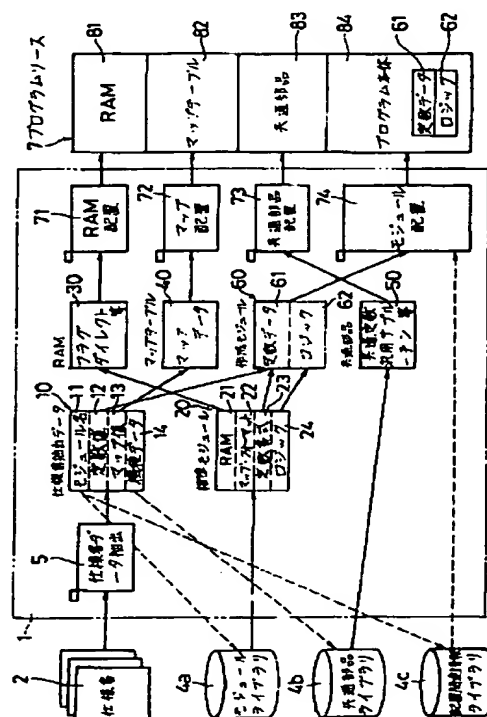
(74)代理人 弁理士 西教 圭一郎

(54) 【発明の名称】 プログラム作成装置

(57) 【要約】

【目的】 プログラム作成を効率的に行い、かつデバッグ作業を容易にする。

【構成】 作成装置 1 は、仕様書 2 から、モジュール名 1 1、定数値 1 2、マップ値 1 3 および機種データ 1 4 などの仕様書抽出データ 1 0 を抽出する。予め登録してあるライブラリ 4 a、4 b、4 c のうち、前記モジュール名 1 1 に対応して、所定の機能を実現するためのプログラムのまとまりである標準モジュール 2 0 およびその標準モジュール 2 0 に関する配置関連情報を抽出するとともに、機種データ 1 4 に対応して掛算や除算等の共通部品 5 0 を抽出する。また、標準モジュール 2 0 中のマップフォーマット 2 2 および定数書式 2 3 に従って、それぞれマップテーブル 4 0 および定数データ 6 1 を作成する。このようにして作成された各部品を、前記配置関連情報に従って配置してプログラムソース 7 を完成する。



【特許請求の範囲】

【請求項 1】 要求されるプログラム仕様に従ってプログラムを作成する装置であって、

予め定める機能単位にまとめられたプログラム部分が標準モジュールとして予め複数登録されるモジュール登録手段と、

各プログラム間で共通に使用可能な共通部品が予め登録される共通部品登録手段と、

前記標準モジュールおよび共通部品を流用するにあたって、該標準モジュールの配置位置や関連するモジュールおよび共通部品を規定する配置関連情報が予め登録される配置関連情報登録手段と、

前記プログラム仕様に対応して、モジュール登録手段から流用すべき標準モジュールを抽出するモジュール作成手段と、

前記プログラム仕様から、使用すべき共通部品を前記共通部品登録手段から抽出する共通部品取出手段と、

前記配置関連情報登録手段から、流用すべき標準モジュールの前記配置関連情報を抽出し、その配置関連情報に基づいた順序で、前記モジュール作成手段において抽出された複数のモジュール、および共通部品取出手段で抽出された共通部品を、複数の各モジュール間で使用されるメモリ領域とともに、作成プログラム中に配置する配置手段とを含むことを特徴とするプログラム作成装置。

【請求項 2】 前記標準モジュールに付随して、該標準モジュール内で使用されるデータの書式が登録されており、

前記モジュール作成手段は、流用すべき標準モジュール内のデータを、前記書式に従って、前記プログラム仕様に対応したデータに書換えることを特徴とする請求項 1 記載のプログラム作成装置。

【請求項 3】 前記データは、演算定数およびマップテーブルにおけるデータポイントの値であることを特徴とする請求項 2 記載のプログラム作成装置。

【請求項 4】 前記プログラム仕様から、作成プログラム中に流用すべき標準モジュールを抽出して、作成プログラム中の対応するモジュールと照合し、前記プログラム仕様に対応した標準モジュールが含まれているか否か、および含まれていて変更が為されているか否かを検証して、その結果を表示する検証手段を備えることを特徴とする請求項 1～3 のいずれかに記載のプログラム作成装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、プログラムを予め定める機能単位にまとめた複数のモジュールに分割し、作成した各モジュールを組合わせて所望とするプログラムを作成するプログラム作成装置に関する。

【0002】

【従来の技術】 従来から、特に大規模なプログラムで

は、その全体を一度に作成するよりも、プログラム全体を、所定の機能のまとまりである複数のブロックに分割し、作成した各ブロック毎のモジュールプログラムを組合わせて、所望とするプログラムに完成させることが多い。特にアセンブル・プログラムは、コンピュータの機械語に対応し、一般にかなり長くなり、このような手法が好適に用いられる。またプログラム作成も、作成期間の短縮化やプログラムの専門分野の細分化などによって、複数人で担当する必要がある。このため、前述のように適度な長さのモジュールに分割してプログラムを作成し、各モジュール毎に動作の確認を行った後に組合わせるようにすれば、プログラムの開発効率が良くなり、動作の信頼性も向上する。

【0003】 典型的な従来技術では、プログラムは、以前に作成したプログラムから、流用すべきモジュールを抽出し、その抽出したモジュールのデータやロジックを、所望とする作成すべきプログラムに要求される仕様に沿って変更および追加し、作成プログラム中に配置することによって新たなプログラムを作成している。

【0004】

【発明が解決しようとする課題】 従来からのモジュール化では、サブルーチンなどの、配置場所に依存しない比較的大きな演算処理ブロックへの区分を行うだけである。また、各モジュールの配置位置は、プログラムが管理するだけである。したがって、各モジュールは任意の箇所に配置することができ、また追加部分を間違った箇所に挿入する可能性があり、バグが発生し易い。

【0005】 また、作成された各モジュールは、ソースプログラムの状態で記述されており、アセンブラやリンカなどと呼ばれるソフトウェアによって、機械語のアセンブル・プログラムに変換される際に組合わせられる。このようなソフトウェアへの指示は、アセンブル・プログラムの文中に予めアセンブラなどの文法に従って記述しておく必要がある。また、アセンブル・プログラム中で定数やマップの形でデータを使用するときには、アセンブラなどの文法に従ってデータに対応するパラメータを与える必要がある。

【0006】 特定の分野に使用されるアセンブル・プログラム、たとえば自動車のエンジン制御のためのプログラムは、車種やグレード毎にデータなどが少しずつ異なるけれども、プログラムの構成自体は殆ど共通である。このため、異なるモジュールについては、複数のモジュールを予め用意しておき、全体のアセンブル・プログラムを作成する際に必要なモジュールを選択する方法で行うことが効率的である。しかしながら、そのようなモジュールの選択は、アセンブル・プログラムのメインルーチンを記述する形で行わなければならないので、新たなプログラムの作成の際や、修正には手間がかかり、さらに自動化することは困難である。

【0007】 さらにまた、アセンブル・プログラム中で

は、データの変更を一定の文法に従って表現するけれども、実際のデータとして書込みが行われるのはオブジェクトコードに変換されるときである。このため、モジュールを組合せたアセンブル・プログラムを見ても、所望とするデータが確実に書込まれているか否かを判断することができず、デバッグ作業に時間がかかるという問題もある。

【0008】本発明の目的は、効率的にプログラムを作成することができるとともに、デバッグ作業も容易なプログラム作成装置を提供することである。

【0009】

【課題を解決するための手段】本発明は、要求されるプログラム仕様に従ってプログラムを作成する装置であって、予め定める機能単位にまとめられたプログラム部分が標準モジュールとして予め複数登録されるモジュール登録手段と、各プログラム間で共通に使用可能な共通部品が予め登録される共通部品登録手段と、前記標準モジュールおよび共通部品を流用するにあたって、該標準モジュールの配置位置や関連するモジュールおよび共通部品を規定する配置関連情報が予め登録される配置関連情報登録手段と、前記プログラム仕様に対応して、モジュール登録手段から流用すべき標準モジュールを抽出するモジュール作成手段と、前記プログラム仕様から、使用すべき共通部品を前記共通部品登録手段から抽出する共通部品取出手段と、前記配置関連情報登録手段から、流用すべき標準モジュールの前記配置関連情報を抽出し、その配置関連情報に基づいた順序で、前記モジュール作成手段において抽出された複数のモジュール、および共通部品取出手段で抽出された共通部品を、複数の各モジュール間で使用されるメモリ領域とともに、作成プログラム中に配置する配置手段とを含むことを特徴とするプログラム作成装置である。

【0010】また本発明は、前記標準モジュールに付随して、該標準モジュール内で使用されるデータの書式が登録されており、前記モジュール作成手段は、流用すべき標準モジュール内のデータを、前記書式に従って、前記プログラム仕様に対応したデータに書換えることを特徴とする。

【0011】さらにまた本発明の前記データは、演算定数およびマップテーブルにおけるデータポイントの値であることを特徴とする請求項2記載のプログラム作成装置。

【0012】また本発明は、前記プログラム仕様から、作成プログラム中に流用すべき標準モジュールを抽出して、作成プログラム中の対応するモジュールと照合し、前記プログラム仕様に対応した標準モジュールが含まれているか否か、および含まれていて変更が為されているか否かを検証して、その結果を表示する検証手段を備えることを特徴とする。

【0013】

【作用】本発明に従えば、プログラムをモジュールと呼ばれる複数の部分に分割してそれぞれ作成し、組合わせて所望とするプログラムへと完成するにあたって、モジュール登録手段には、前記各部分の原型となる標準モジュールが予め複数登録されている。前記標準モジュールは、作成すべきプログラムが、たとえば自動車の電子制御装置のためのものであるときには、各センサからの出力の読み込み、マップデータの読出し、制御量演算およびアクチュエータへの制御出力などのように、比較的細かな、予め定める機能単位にまとめられて登録されている。

【0014】これに対応して、前記標準モジュールの配置位置に依存する配置関連情報が、配置関連情報登録手段に登録されている。前記配置関連情報は、たとえば標準モジュールが掛算を含むモジュールである場合、その標準モジュールに近接して掛算のルーチンを配置すべきであるなどの、各標準モジュールのプログラム中での関連と、他の関連モジュールとの相対位置を規定する情報である。

【0015】また、前記掛算の計算手順などの複数の各標準モジュール間で共通に使用可能な共通部品が、共通部品登録手段に予め登録されている。この共通部品に関連するモジュールおよびそれとの相対的な配置位置を規定する配置関連情報も、前記配置関連情報登録手段に登録されている。

【0016】このようにして、作成プログラムの原型となる各部分およびそれに関連する情報が登録されており、モジュール作成手段は、前記プログラム仕様から、流用すべき標準モジュールをモジュール登録手段から抽出する。各標準モジュールは、たとえば演算処理手順を表すロジック部分と、演算定数や、マップテーブルにおけるデータポイントの値などのデータ部分とを含んで構成されている。

【0017】また好ましくは、それらのデータの書式も併せて構成されている。モジュール作成手段は、上述のように流用すべき標準モジュールを抽出すると、その標準モジュール内のデータを前記書式に従って、プログラム仕様に対応したデータに書換えてゆく。こうして、前記演算定数やマップ値などが所望とする書式で自動的に書換えられてゆく。

【0018】また、前記プログラム仕様から、共通部品登録手段に登録されている共通部品のうち、使用すべき共通部品が、共通部品取出手段によって抽出される。このようにして、使用すべき標準モジュールおよび共通部品が取出されると、配置手段は、抽出されている標準モジュールに対応した配置関連情報を配置関連情報登録手段から抽出し、その配置関連情報に基づいた順序でモジュールおよび共通部品を作成プログラム中に配置してゆく。このときまた配置手段は、複数の各モジュール間で使用される、たとえば変数やフラグの定義などに使用さ

れるメモリ領域も配置して、所望とする作成プログラムが完成する。

【0019】したがって、プログラムの構成自体には変化がなく、プログラムの仕様に従ってデータのみを変更するようなどときには、既存の標準モジュールや共通部品を適所に配置し、かつデータの更新も行つて自動的に所望とする作成プログラムを完成することができる。したがって、新たに標準モジュールや共通部品を作成して登録しておく必要はなく、また各モジュールの配置ミスなどによるバグの発生の恐れも少なく、効率的にプログラムを作成することができるとともに、デバッグ作業を軽減することができる。

【0020】好ましくは、プログラム仕様から、作成プログラム中に流用すべき標準モジュールを抽出して、作成プログラム中の対応するモジュールと照合する検証手段を設ける。この検証手段は、作成プログラム中に、前記プログラム仕様に対応した標準モジュールが含まれているか否か、および含まれていて変更が為されているか否かなどを検証してその結果を表示する。したがって、プログラム仕様に対応したデータなどが作成プログラムに反映されているかなどを容易に確認することができる。

【0021】

【実施例】図1は、本発明の一実施例のプログラムの作成手順の概略を示す図である。エンジニアリングワークステーションなどで実現される作成装置1は、仕様書2に基づいて、所望とする作成プログラム3を完成させる。前記仕様書2には、作成プログラム3が、たとえば自動車のエンジン制御用であるときには、そのエンジンの特性に対応した制御定数や制御マップなどの要求されるスペックを表すデータが、車種名やエンジンの形式名等の機種名などとともに電子化データで記載されている。

【0022】前記作成装置1は、仕様書2の内容を読み込み、仕様書データ抽出部5において仕様書2中に記載された抽出データ10を得て、プログラム自動作成部6に与える。プログラム自動作成部6は、モジュール名や機種情報などの前記抽出データ10に対応して、予めライブラリ4に登録されている複数のプログラムソースから流用できそうなプログラムソースを選択し、そのプログラムソースの内容を前記抽出データ10に従って書換えてゆく。こうして得られた各プログラムソースが組合せられて、プログラムソース7が作成される。

【0023】前記プログラムソース7は、修正部8において、プログラマが陰極線管などの表示手段18を参照しながらキーボード17から入力操作を行うことによって、追加や変更が加えられる。こうして作成された作成プログラム3は、仕様データ照合部9において、前記抽出データ10と照合され、仕様書2の内容との一致点や相違点が前記表示手段18によって表示され、必要が

あれば再び前記修正部8において修正されて完成される。

【0024】図2は、上述の作成手順を詳細に示すブロック図である。なおこの図2では、アセンブル・プログラムを作成するものとし、また前述の図1におけるプログラムソース7の作成までを示す。作成装置1では、作成プログラム3が前述のように自動車のエンジン制御用であるときには、たとえば吸気圧読み込み、回転数読み込み、燃料噴射出力および点火出力などの比較的細かな機能毎のまとまりであるプログラム部分をモジュールとして登録しておき、仕様書2で要求される仕様に従って、それらのモジュールを選択的に使用して書換えなどを行って組合わせることによって、所望とする作成プログラム3に完成させる。したがって、仕様書2から仕様データ抽出部5で抽出された抽出データ10には、使用するべき各モジュール毎に、モジュール名11と、データである定数値12およびマップ値13と、機種データ14とが含まれている。

【0025】前記モジュール名11は、前述のような吸気圧読み込みおよび回転数読み込みなどのそのモジュールの機能などに対応して付されている名称である。定数値12やマップ値13は、前記各モジュールでの制御定数であり、それらを表すコメント文で区画されて記載されているだけではなく、プログラムの文章中にも記載されることがある。前記定数値12は、たとえばセンサ出力のLSBを表す値などであり、この場合には、前記センサ出力が該定数値で除算されて、実際に制御量演算に用いる値に変換される。

【0026】前記マップ値13は、多次元のデータであり、図3(1)で示すようなマップから、等間隔または不等間隔にデータポイントP1、P2、…を取出して、その座標値を図3(2)で示すような表に表したものであり、前記作成プログラム3を用いる制御装置は、たとえば前記データポイントP2、P3間で、座標値Xaに対応する座標値Yaを前記データポイントP2、P3を用いて補間演算を行って求める。

【0027】前記機種データ14は、たとえば作成プログラム3が上述のようなエンジン制御用であるときには、車種やエンジン型式などの該作成プログラム3を使用すべき装置の名称や型式を表し、これによって、たとえば変速機が自動であるのか手動であるのかなどの、同じモジュール名であっても、後述する使用するべき共通部品50が異なることもあり、そのような選択のために用いられる。

【0028】一方、前記ライブラリ4は、3つのライブラリ4a、4b、4cから構成されている。モジュールライブラリ4aには、以前に作成されたプログラムから抽出された複数のモジュールが、作成プログラム3に用いる原型モジュールとなる標準モジュール20として登録されている。

【0029】また、共通部品ライブラリ4bには、前記仕様書2には記述されていないけれども、前記モジュールライブラリ4aに登録されている各標準モジュール20から共通に使用される可能性のある、たとえば掛算や除算のサブルーチンなどの共通部品50が登録されている。

【0030】配置関連情報ライブラリ4cには、各モジュールの関連、たとえば仕様書2に対応した或るモジュールを選択すると、必ず必要となるモジュール、たとえば選択されたモジュール中に掛算処理が生まれているときには、掛算のモジュールを配置する必要があること、およびそれらの相互の配置位置などの、或るモジュールに対して、関連するモジュールと、その或るモジュールのプログラム上での配置位置などに関連した情報が登録されている。

【0031】これら各ライブラリ4a~4cを用いてプログラムを作成するにあたって、作成装置1は、仕様書抽出データ10のうち、先ずモジュール名11に基づいてモジュールライブラリ4aから標準モジュール20を読み出すとともに、その読み出した標準モジュール20に対応する配置関連情報を配置関連情報ライブラリ4cから読み出す。読み出された標準モジュール20は、前記モジュール名11に対応した、たとえば吸気圧読み込みや、回転数読み込みの動作を実現するためのプログラム部分であり、大略的に、RAM21と、マップ・フォーマット22と、定数書式23と、ロジック24とを含んで構成されている。

【0032】RAM21には、後述するようにその標準モジュール20が必要とする書換え可能メモリを確保するためのアセンブル定義式が記述されている。マップ・フォーマット22には、作成すべきプログラム中で使用されるマップデータの書込みのために必要なフォーマットが記述されている。このマップ・フォーマットの例を図3(3)で示し、この例中において下線部分が前記仕様書2のマップ値13に書換えられることになる。定数書式23には、アセンブル・プログラム中に定数として込むべきデータが記述されている。ロジック24には、その標準モジュール20における演算処理手順がソースコードで記述されている。このような標準モジュール20は、モジュール名11に対応して順次抽出されてゆき、仕様書抽出データ10に基づいて以下のように書換えが行われてゆく。

【0033】先ず、図3(2)で示されるような仕様書抽出データ10内のマップ値13が図3(3)で示す標準モジュール20内のマップ・フォーマット22に従って記述されてゆく、図3(4)で示すようなマップテーブル40のためのマップデータが作成される。

【0034】この場合、図3(1)において参照符P1、P2、...で示すデータポイントの数が、図3(3)で示す標準モジュール20のマップ・フォーマット22

では2つであるのに対して、図3(2)で示す仕様書2のマップ値13では3つであり、このようにデータポイントの数が相互に異なるときには、仕様書2のマップ値13におけるデータポイント数だけマップ・フォーマットが複製されて、図3(4)で示すように仕様書2のマップ値13に対応した数だけデータポイントが設定される。

【0035】またこのとき、図3(3)で示すような小数点データと、図3(2)で示すような整数データとのようにデータの型式が不一致であるときには、前記図1の仕様書データ照合部9から表示手段18に警告表示が行われる。

【0036】同様に、仕様書抽出データ10の定数値12は、標準モジュール20の定数書式23に従って書換えられる。すなわち、前記定数値12が、たとえばAAAという変数を20に設定する、

AAA = 20

と記載されているとき、定数書式23が、

AAA EQU 10.5 * 256 / 4

であるときには、下線部分に定数値12の「20」が込まれるので、

AAA EQU 20 * 256 / 4

が作成された定数データ61となる。なお、この場合もデータの型式が相互に異なるときには、上述と同様に警告が発せられる。

【0037】こうして作成された定数データ61は、標準モジュール20のロジック24を移植して作成されたロジック62とともに、作成モジュール60としてモジュール配置部74で以下のように配置が行われる。

【0038】仕様書抽出データ10のモジュール名11に基づいて配置関連情報ライブラリ4cから読み出される配置関連情報63は、図4で示すように記述されている。この配置関連情報63は、関連情報64と、配置情報65とを含んで構成されている。

【0039】この図4では、関連情報64aは、Xという名称の標準モジュールを使用すると、Cという名称の標準モジュールも使用する必要があり、同様に関連情報64bは、標準モジュールCを使用すると標準モジュールXを使用する必要があることを表す。また配置情報65からは、使用される標準モジュールは、BEGIN、END、X、A、D、BおよびEであることが理解される。したがって、標準モジュールXが使用されるので、前記関連情報64aから、標準モジュールCも使用されることとなる。

【0040】また配置情報65aからは、トという記号は、その記号トが付された標準モジュールBEGINが、プログラム本体の先頭に配置されるべきことを表し、配置情報65bからは、記号トは、標準モジュールENDが前記プログラム本体の最後に配置されるべきことを表す。また、配置情報65cからは、記号<は、標

標準モジュールXが標準モジュールAの前に配置されるべきことを表し、配置情報65dからは、記号<は、標準モジュールBが標準モジュールEの直前に配置されるべきことを表す。なお、標準モジュールCに対する配置情報は何も記述されていないので、この標準モジュールCは上述の結果から、プログラム本体の最初と最後とを除いて、どこに配置されてもよい。

【0041】図5は、上述のようなモジュール配置関連情報63に従う前記モジュール配置部74によるモジュール配置手順の考え方を説明するためのブロック図である。まず、仕様書抽出データ10のモジュール名11に対応して、配置関連情報ライブラリ4cのライブラリ66から関連情報が読出され、すなわち前記図4における関連情報64bに対応したモジュールXも前記モジュール名11に加えられる。次に、こうして求められた配置要求のある全てのモジュール名に対応して、モジュールライブラリ4aから標準モジュール20が読出される。続いて、これらの各標準モジュール20の配置情報が、配置関連情報ライブラリ4cのライブラリ67から読出され、それによって配列されてプログラム本体84が作成されて、図2で示すようにプログラムソース7内に配置される。

【0042】図6は、共通部品ライブラリ4bに登録される共通部品50の種類の一例を示す図である。この共通部品ライブラリ4bに登録されている共通部品50は、仕様書2には表れていないけれども、各標準モジュールから共通に使用される可能性のある共通定数や、汎用サブルーチン等から成る。この共通部品50は、以前に作成されたプログラムから抽出されて、機種データとともに前記共通部品ライブラリ4bに登録されている。したがって、仕様書抽出データ10の機種データ14に対応して、必要な共通部品50が共通部品ライブラリ4bから読出される。

【0043】前記共通部品50は、具体的には、この図6で示すように、前記共通定数の定義文、マクロ定義文、レジスタ定義文、汎用サブルーチンおよび初期化ルーチンなどである。

【0044】共通定数の定義文とは、プログラム中で使用するデータを定義するためのものであり、直接的にデータを演算処理のための値に定義するのが困難なときに使用され、たとえば冷却水温度は、プログラム中では変数THWで定義され、その変数THWの実際の値は、後述するプログラムソース7内のRAM81内に記述されることになる。

【0045】また、マクロ定義文は、プログラマが複数の命令を組合わせて任意に作成した大きな命令を表し、たとえば「冷却水温度の読み込み」とマクロ定義されている場合、まず複数の温度センサの出力が与えられるアナログ/デジタル変換器への入力を冷却水温度センサの出力に切り換え、次に入力されたセンサ出力のアナログ/デ

ジタル変換を行い、続いてその変換結果をレジスタに読込むというものである。

【0046】さらにまた、レジスタ定義文は、マイクロコンピュータのRAM中でレジスタとなるべきアドレスを定義するためのものであり、またそのレジスタの論理に対するポートの論理の定義などを行うことができる。汎用サブルーチンは、前述のように4則演算などであり、初期化ルーチンは、フラグやカウンタなどの初期化を行うためのものである。

【0047】このような共通部品50は、共通部品ライブラリ4bに登録されている共通部品の、図7で示すようなそのリストに、前記仕様書抽出データ10内の機種データ14およびキーボード17からの入力が対照されて抽出される。この図7で示すリスト中で、部品種類No.は、前記図6において①、②、…を付して示す定義文やルーチンの識別番号であり、機種名は、前記機種データ14に対応しており、共通部品ファイル名は、該共通部品ライブラリ4b中に登録されている前記定義文やルーチンの名称である。こうして抽出された共通部品50は、共通部品配置部73によってプログラムソース7内に、参照符83で示すように配置される。前記プログラムソース7内にはまた、マップテーブル40が、マップ配置部72によって参照符82で示すように配置される。

【0048】モジュールライブラリ4aから抽出された標準モジュール20には、メモリ領域であるRAM21が定義されている。これは、本発明で用いるアセンブラ言語では、プログラム中で使用する変数を抽出して一括して同じ場所に定義する必要があるためである。RAM21は、フラグ領域と、ダイレクト領域と、エクステンデッド領域と、スタンバイ領域となどの各領域が必要とする量だけ割付けられて構成されている。

【0049】前記フラグ領域は、フラグとして使用できるRAMの位置をビット単位で定義することができる領域である。前記ダイレクト領域は、たとえば「0033」などのアドレスを「33」という具合に、1バイトの簡略化したデータでアドレス指定可能な、いわゆるショートアドレッシング領域である。これに対してエクステンデッド領域は、2バイトのデータでアドレス指定を行う、いわゆるフルアドレッシング領域である。さらにスタンバイ領域は、プログラム停止時もデータが保存されるべき不揮発性の領域である。

【0050】図8は、前記RAM21の内容の一例を示す図である。アセンブル・プログラム中では、変数は変数名で記述されており、後述するようにソフトウェアを用いてプログラム展開すると前記変数名はアドレスとそれをコールする命令とに変換され、前記変数名とアドレスとを対応付けるのがこのRAM21である。

【0051】前記RAM21において、参照符91で示す部分はフラグ領域であることを表し、参照符92で示

す部分は、「A」という名称のフラグがこのフラグ領域の1ビット目に定義されていることを表す。また参照符93で示される部分はダイレクト領域であることを表し、参照符94、95で示す部分は、それぞれ「B」および「C」という名称の変数に、それぞれ2バイトと、1バイトとの領域が割当てられていることを表す。

【0052】このようなRAM21を有する各標準モジュール20がプログラム作成のために前述のように順次読出されてゆくと、前記図2において参照符30で示すように、フラグ領域やダイレクト領域などがそれぞれ重複しないように集約されたRAMが形成される。

【0053】すなわち、たとえば図9(1)で示すような各種別毎のそれぞれビット位置またはデータ長を有するRAM96と、図9(2)で示すような各種別毎のビット位置またはデータ長を有するRAM97とが組合わせられると、図9(3)のRAM98で示すようになる。したがって、たとえばフラグ領域ではRAM96のフラグAと、RAM97のフラグKとがそれぞれRAM98において割付けられており、またRAM96では2バイトで割当てられているダイレクト領域Bと、RAM97では1バイトで割当てられているダイレクト領域Bとは統合されて、RAM98では2バイトのダイレクト領域Bとなっている。

【0054】このようなRAM30の割付けがRAM配置部71で行われて、プログラムソース7内に参照符81で示すように配置される。こうして、プログラムソース7が完成する。

【0055】このようにして作成されたソースプログラム7は、前述の図1で示されるように修正部8において追加や変更が加えられ、作成プログラム3が完成する。この作成プログラム3は、たとえばその70%程度がプログラムソース7を流用し、修正手段8において残余の30%程度が新規に追加や変更されて作成されている。この作成プログラム3は、図1および図2を参照して、以下のような手順で仕様書データ照合部9において照合が行われる。

【0056】仕様書データ照合部9は、まず仕様書抽出データ10を読み、モジュール名11と、機種データ14とに基づいて、ライブラリ4から前記標準モジュール20および共通部品50を読出す。また、作成プログラム3のプログラム本体84から、コメント文で記述されているモジュール名を読出す。なお、各モジュールはその前後に区切りを示すキーワードが挿入されており、このキーワードに従って各モジュールが抽出され、モジュール名が抽出される。その抽出されたモジュール名と、仕様書抽出データ10のモジュール名11とが比較され、一致していないときには、表示手段18によって変更などが加えられていることを表示する。

【0057】次に、各モジュール単位での定数データ61やマップテーブル82の比較が行われる。このアセン

ブル・プログラムでは、プログラム中にはEQUで示される定義文によって定数を定義する疑似命令が挿入されており、前記比較は、こうして定義された定数がまとめて記述されている部分同士で行われ、変更が為されているときには、上述のようにそのことを表す表示が行われる。プログラマは、このような表示を参照して、デバッグ作業を行う。

【0058】このように本発明に従う作成装置1では、上述の実施例のようなエンジン制御用の他に、自動変速機の制御やブレーキの制御などの他の制御用にもモジュールの共用化を可能とするために、所定の機能毎に細分化した標準モジュール20を自動的に組合わせるとともに、作成すべきプログラムの仕様に従って自動的にデータをプログラム中に書き込むことによってプログラムソース7を作成するので、プログラムソース7を容易に作成することができ、開発効率を向上することができる。

【0059】また、各標準モジュール20の前記プログラムソース7内での配置位置および各標準モジュール20に関連する共通部品50などの情報を配置関連情報ライブラリ4c内に登録しておき、その情報を用いてモジュール配置を行うので、各標準モジュール20はそれぞれの配置情報および関連情報の制約の範囲内でしか配置されないで、モジュールの配置をもれなく、正確に行うことができ、バグの発生を抑えて、プログラムの信頼性を向上することができる。

【0060】さらにまた、標準モジュール20のデータの記述形式をライブラリ化し、仕様書抽出データ10を作成プログラム3に書き込むことによって、人手による入力ミスを防いで、仕様書2のデータを確実にプログラムに反映させることができ、プログラムの信頼性をプログラム作成の段階で確保できる。

【0061】また、プログラム間で共通使用可能な共通部品50を、検索情報である機種データ14とともにライブラリ化することによって、プログラム仕様の情報に基づいた共通部品50の自動配置が実現でき、プログラム作成の工数をさらに削減できる。

【0062】さらにまた、プログラム仕様から抽出されるデータと作成プログラム中に書込まれたデータとの一致を検証することができるので、プログラムのデバッグ作業が容易にできる。

【0063】また、作成プログラム3をソースプログラムの状態で各標準モジュール20や共通部品50と照合するので、照合作業が容易である。さらにまた、仕様2に記載されたデータと、作成プログラム3に書込まれたデータとが一致しているか否かを検証することができるので、データを確実にプログラムに反映させることができる。

【0064】なお、上述の実施例で作成された作成プログラム3は、ソースコードで記述されたアセンブル・プログラムであり、アセンブラなどのソフトウェアを用い

ることによって、オブジェクトコードに変換することができる。また、本発明の他の実施例として他の言語が用いられてもよく、さらに本発明ではハードウェアとして作成装置 1 に、いわゆるエンジニアリングワークステーションを用いているけれども、パーソナルコンピュータなどの他の種類のコンピュータを用いても良いことは勿論である。

【0065】

【発明の効果】以上のように本発明によれば、細分化したモジュールをそれぞれ作成して組合わせることによって、所望とするプログラムに完成するにあたって、各モジュールに配置位置に関連する情報を持たせ、その配置関連情報に基づいて、各モジュールやそれらに用いられるべき共通部品を配置するので、プログラム仕様に沿った所望とするプログラムを効率的に作成することができる。また、バグの発生を抑えてデバッグ作業を軽減することができる。

【0066】好ましくは、プログラム仕様から、作成プログラム中に流用すべき標準モジュールを抽出して、作成プログラム中の対応するモジュールと照合する検証手段を設ける。この検証手段は、作成プログラム中に前記プログラム仕様に対応した標準モジュールが含まれているか否か、および含まれていて変更が為されているか否かなどを検証してその結果を表示する。したがって、プログラム仕様に対応したデータなどが作成プログラムに反映されているかなどを容易に確認することができる。

【図面の簡単な説明】

【図 1】本発明の一実施例のプログラムの作成手順の概略を示す図である。

【図 2】図 1 の作成手順を詳細に示すブロック図である。

【図 3】マップテーブル 40 の書換手順を説明するため

の図である。

【図 4】モジュール配置関連情報 63 の一例を示す図である。

【図 5】モジュール配置手順の考え方を説明するためのブロック図である。

【図 6】複数の各標準モジュール間で使用される可能性のある共通部品 50 の一例を示す図である。

【図 7】図 6 の共通部品 50 のリストの一例を示す図である。

【図 8】各標準モジュール 20 内の RAM 21 の内容の一例を示す図である。

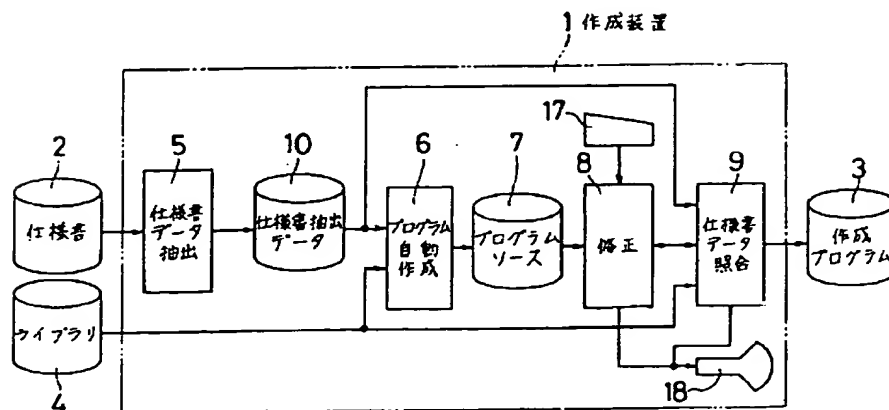
【図 9】RAM 配置部 71 の動作を説明するための図である。

【符号の説明】

- 1 作成装置
- 2 仕様書
- 3 作成プログラム
- 4 a モジュールライブラリ
- 4 b 共通部品ライブラリ
- 4 c 配置関連情報ライブラリ
- 5 仕様書データ抽出部
- 6 プログラム自動作成部
- 7 プログラムソース
- 8 修正部
- 9 仕様書データ照合部
- 10 仕様書抽出データ
- 20 標準モジュール
- 30 RAM
- 40 マップテーブル
- 50 共通部品
- 60 作成モジュール
- 71～74 配置部

【図 1】

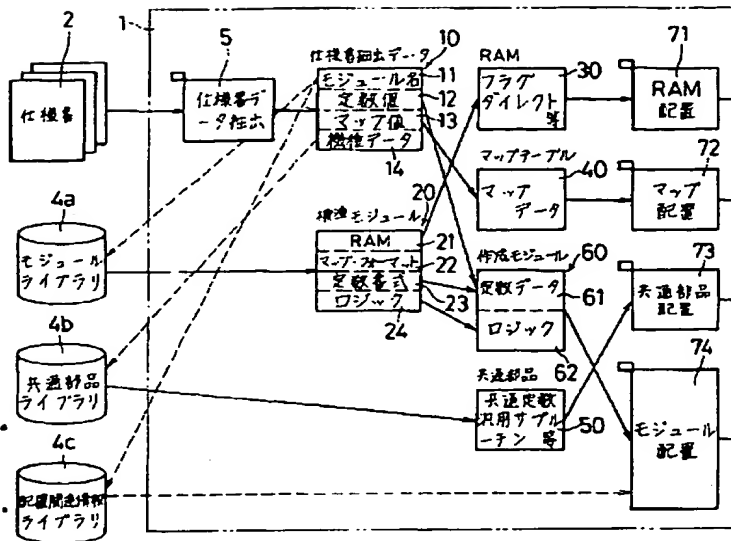
【図 6】



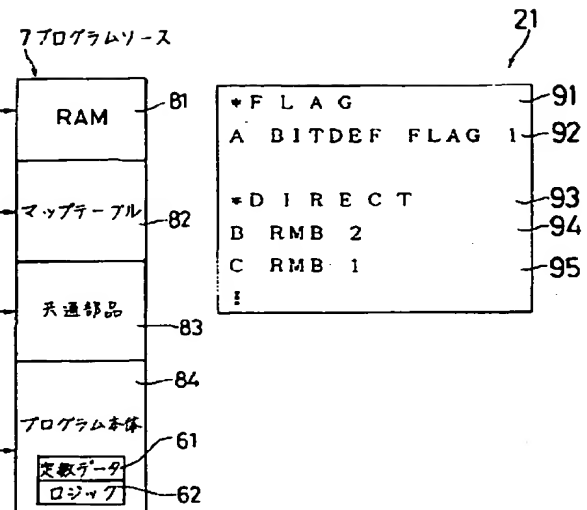
- ①共通定数定義
- ②マクロ定義
- ③レジスタ定義
- ④汎用サブルーチン
- ⑤初期化ルーチン

50

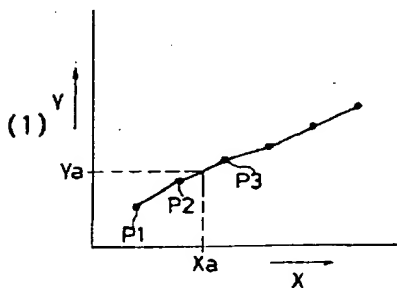
【图2】



【图 3】



【図 4】



(2)

仕様書マップ値13			
X	1	2	3
Y	100	200	300

マップフォーマット22、

(3) MAP TABL
FCB 1.00/60, 111*256/4
FCB 2.00/60, 222*256/4
MAPX EQU *

プログラムのマップテーブル40、

(4) MAP TABL
FCB 1/60. 100*256/4
FCB 2/60. 200*256/4
FCB 3/60. 300*256/4
MAPX EQU *

The diagram shows a program listing with two sections, each enclosed in a rectangular box. The first section is labeled '#RELATION START' and contains two lines: 'モジュールX [モジュールC]' and 'モジュールC [モジュールX]'. The second section is labeled '#RELATIONE END'. To the right of the first section, a bracket groups the two lines and is labeled '64'. The second section is labeled '64a' for the first line and '64b' for the second line. Below the first section, there is another rectangular box labeled '#SEQ START'. It contains four lines: 'ト モジュール BEGIN', 'イ モジュール END', 'モジュールX < モジュールA', and 'モジュールD < モジュールA'. To the right of this box, a bracket groups the last three lines and is labeled '65'. The lines are individually labeled: '65a' for 'ト モジュール BEGIN', '65b' for 'イ モジュール END', '65c' for 'モジュールX < モジュールA', and '65d' for 'モジュールD < モジュールA'. Below the second box, there is a third rectangular box labeled '#SEQ END'.

```

#RELATION START
モジュールX [モジュールC]
モジュールC [モジュールX]
#RELATIONE END

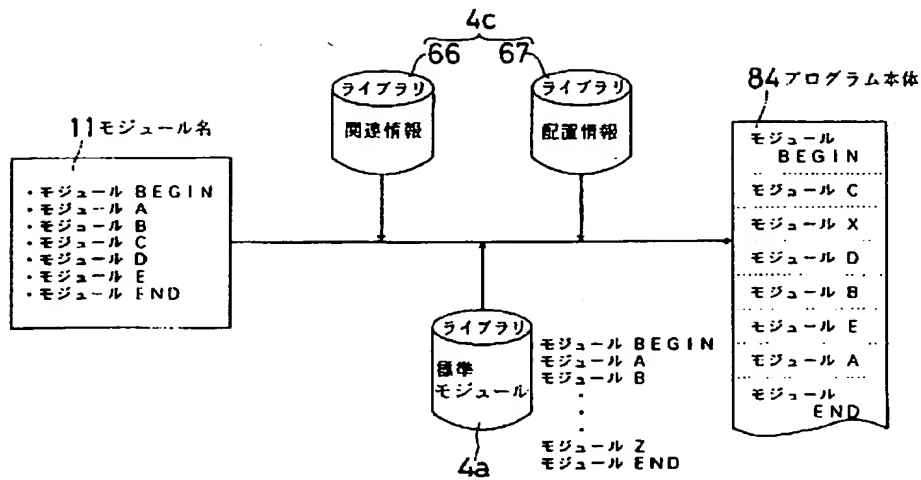
#SEQ START
ト モジュール BEGIN
イ モジュール END
モジュールX < モジュールA
モジュールD < モジュールA
モジュールB |< モジュールE
モジュールE < モジュールA
#SEQ END

```

【圖 7】

部品種類 No.	機種名	共通部品 ファイル名
1	○○○-○○	***
2	△△△-○○	※※※
2	△△△-○△	
2	△△△-△△	
3	□□□-○○	
⋮		

【図5】



【図9】

(1)

変数名	種 別	ビット位置 データ長
A	フラグ	1 (ビットの1番目)
B	ダイレクト	2 (バイト数)
C	エクステンデ	1
D	エクステンデ	2
E	スタンバイ	2
F	ダイレクト	1

96

(2)

変数名	種 別	ビット位置 データ長
K	フラグ	5
B	ダイレクト	1
F	ダイレクト	1
G	エクステンデ	1
H	エクステンデ	2

97

(3)

フラグ	A	1
	K	5
ダイレクト	B	2
	F	1
エクステンデ	C	1
	D	2
	G	1
	H	2
スタンバイ	E	2

98